eSecurity

Treinamentos em Segurança da Informação

O que temos para hoje?

Menu do dia:

- ✓ **Auditoria de Senhas:**
  - ✓ O protocolo HTTP
  - ✓ Cookies
  - ✓ Sequestro de sessão
  - ✓ Exemplo 1: Usando Sniffing
  - ✓ Exemplo 2: Usando XSS
  - ✓ Exemplo 3: Prediction Attack

Auditoria Passiva de Senhas

Sequestro de Sessão

O protocolo HTTP é stateless, ou seja, ele não mantém um estado/conexão. Toda a interação que o seu cliente fizer com um servidor Web acarretará em uma nova requisição e resposta.

As requisições são independentes e possuem um tempo de vida (conexão, envio de mensagem, resposta, encerramento da conexão). O servidor Web não é capaz de identificar se duas requisições vieram de um mesmo navegador, e o mesmo não faz nenhum gerenciamento em memória para que mensagens sejam compartilhadas entre requisições.

É para suprir esta necessidade que entram os cookies e sessões.

Os Cookies

Através de cookies o servidor Web é capaz de trocar informações de estado com o navegador do usuário. Desse modo, somos capazes de adicionar produtos a um carrinho de compras, sem perder estas informações ao mudar de página, sair do website ou até mesmo fechar o navegador.

Tecnicamente falando, um cookie é uma pequena quantidade de informação persistida temporariamente pelo navegador. Os navegadores normalmente limitam o tamanho dos cookies em até 4KB, e apagam cookies com a data de "validade vencida".

Fonte: klauslaube

Sequestro de Sessão

Em ciência da computação, sequestro de sessão, algumas vezes também conhecido como sequestro de cookie é a exploração de uma sessão de computador válida, às vezes também chamada de uma chave de sessão - para obter acesso não autorizado a informações ou serviços em um sistema de computador.

Particularmente é utilizado para se referir ao roubo de um cookie, utilizado para autenticar um usuário em um servidor remoto. Ele tem particular relevância para os desenvolvedores web, quando os cookies HTTP usados para manter uma sessão em vários sites web podem ser facilmente roubados por um atacante (cracker) usando um computador intermediário ou com acesso aos cookies guardados no computador da vítima.

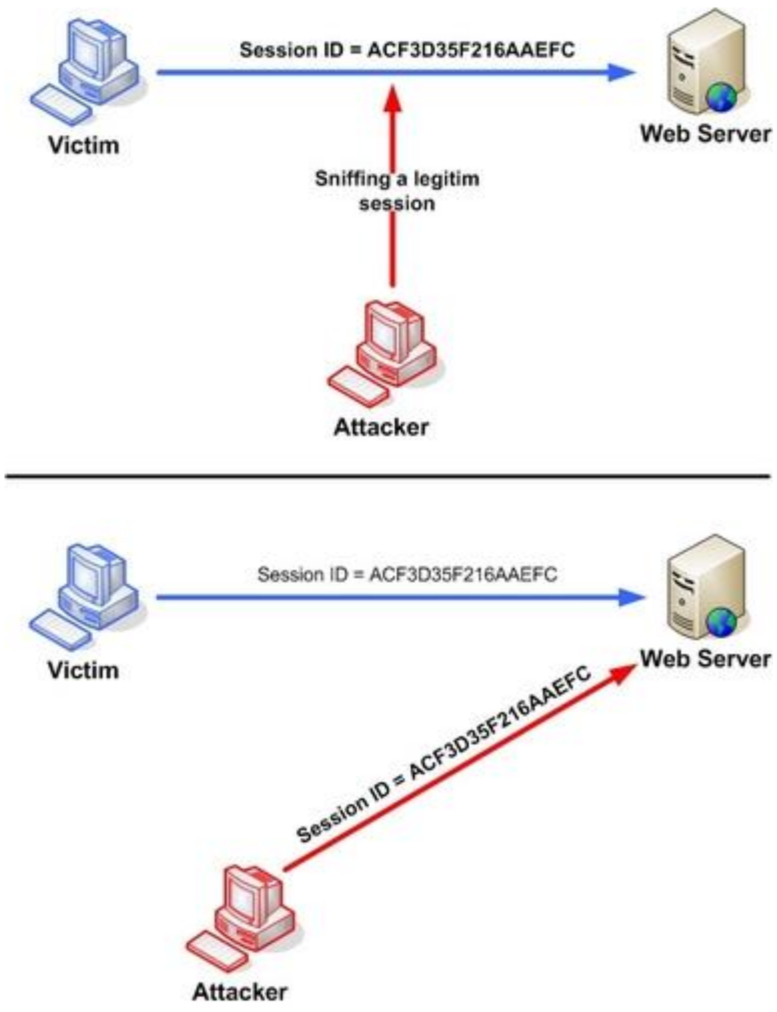
Um método popular é usar pacotes IP de origem roteada. Ou seja, um sniffer.

Se o roteamento de origem for desligado, o cracker pode usar sequestro "cego", pelo que adivinha as respostas das duas máquinas. Assim, o cracker pode enviar um comando, mas nunca pode ver a resposta.

Fonte: Wiki

Sequestro de Sessão: Exemplo 1

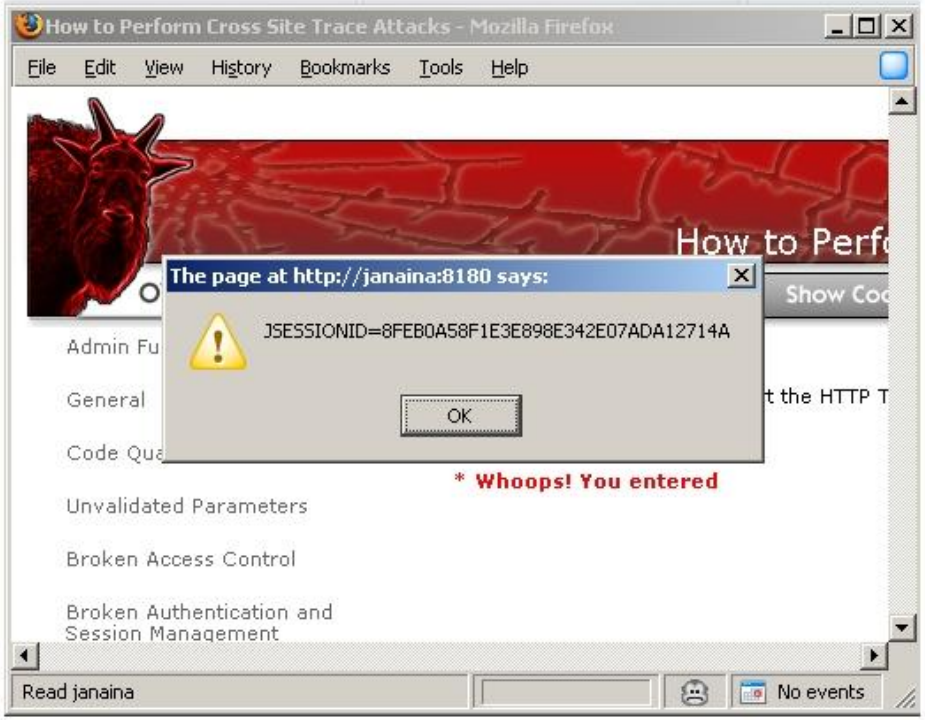
No exemplo, como podemos ver, em primeiro lugar o atacante usa um sniffer para capturar uma sessão de token válido chamado "Session ID", então ele usa a sessão token válido para obter acesso não autorizado ao servidor Web



Sequestro de Sessão: Exemplo 2

Neste caso, o atacante pode comprometer o token de sessão por meio de código malicioso ou programas em execução ao lado do cliente.

O exemplo mostra como o invasor pode usar um ataque XSS para roubar o token de sessão. Se um invasor envia um link criado para a vítima com o JavaScript malicioso, quando a vítima clica no link, o JavaScript será executado e seguirá as instruções criadas pelo atacante.



Exemplo: <SCRIPT>alert(document.cookie);</SCRIPT>

Sequestro de Sessão: Exemplo 3

O ataque previsão sessão, possui a previsão de valores do ID da sessão que permite que um invasor contorne o esquema de autenticação de um aplicativo.

Ao analisar e compreender o processo de geração de ID da sessão, um atacante pode prever um valor SESSION ID válido e adquira acesso ao aplicativo.

Na primeira etapa, o atacante precisa coletar alguns valores SESSION ID válidos que são usados para identificar os usuários autenticados. Em seguida, ele deve entender a estrutura da sessão, a informação que é usado para criá-lo, e o algoritmo de criptografia ou hash usado pelo aplicativo para protegê-lo. Algumas implementações ruins são compostas por nome de usuário ou outras informações previsíveis, como timestamp ou endereço IP. No pior dos casos, essas informações são usadas em texto simples ou codificados com um algoritmo como a codificação base64.

Além disso, o atacante pode implementar uma técnica de força bruta para gerar e testar diferentes valores de sessão até que ele receba com êxito o acesso ao aplicativo.

Chega por hoje

www.eSecurity.com.br

E-mail: alan.sanches@esecurity.com.br

Twitter: @esecuritybr e @desafiohacker

Skype: desafiohacker

Fanpage: www.facebook.com/academiahacker